LEDAkem: a post-quantum key encapsulation mechanism based on QC-LDPC codes

Marco Baldi[1], Alessandro Barenghi[2], Franco Chiaraluce[1], Gerardo Pelosi[2], Paolo Santini[1]

[1]Università Politecnica delle Marche
(m.baldi@univpm.it, f.chiaraluce@univpm.it, p.santini@pm.univpm.it)
[2]Politecnico di Milano
(alessandro.barenghi@polimi.it, gerardo.pelosi@polimi.it)

PQCrypto 2018
The Ninth International Conference on Post-Quantum Cryptography

Fort Lauderdale, Florida April 9-11, 2018

Code-based crypto

- Code-based public-key cryptosystems were introduced by McEliece in 1978, exploiting Goppa codes.
- Besides quantum resistant, they are algorithmically efficient.

▶ R. McEliece, "Public-Key System Based on Algebraic Coding Theory," DSN Progress Report 44, pp. 114–116, 1978.

Code-based crypto

- Code-based public-key cryptosystems were introduced by McEliece in 1978, exploiting Goppa codes.
- Besides quantum resistant, they are algorithmically efficient.
- In 1986 Niederreiter introduced a variant in the syndrome domain, while McEliece works in the codeword domain.

- R. McEliece, "Public-Key System Based on Algebraic Coding Theory," DSN Progress Report 44, pp. 114–116, 1978.
- H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," Problems of Control and Information Theory, vol. 15, pp. 159–166, 1986.

Code-based crypto

- Code-based public-key cryptosystems were introduced by McEliece in 1978, exploiting Goppa codes.
- Besides quantum resistant, they are algorithmically efficient.
- In 1986 Niederreiter introduced a variant in the syndrome domain, while McEliece works in the codeword domain.
- McEliece and Niederreiter indeed are two formulations of the same code-based trapdoor.

- R. McEliece, "Public-Key System Based on Algebraic Coding Theory," DSN Progress Report 44, pp. 114–116, 1978.
- H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," Problems of Control and Information Theory, vol. 15, pp. 159–166, 1986.
- Y. X. Li, R. H. Deng and X. M. Wang, "On the equivalence of McEliece's and Niederreiter's public-key cryptosystems," IEEE Trans. Inf. Theory, vol. 40, no. 1, pp. 271–273, Jan 1994.

Code-based crypto

- Code-based public-key cryptosystems were introduced by McEliece in 1978, exploiting Goppa codes.
- Besides quantum resistant, they are algorithmically efficient.
- In 1986 Niederreiter introduced a variant in the syndrome domain, while McEliece works in the codeword domain.
- McEliece and Niederreiter indeed are two formulations of the same code-based trapdoor.
- Goppa codes have resisted cryptanalysis for 40 years...

- R. McEliece, "Public-Key System Based on Algebraic Coding Theory," DSN Progress Report 44, pp. 114–116, 1978.
- H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," Problems of Control and Information Theory, vol. 15, pp. 159–166, 1986.
- Y. X. Li, R. H. Deng and X. M. Wang, "On the equivalence of McEliece's and Niederreiter's public-key cryptosystems," IEEE Trans. Inf. Theory, vol. 40, no. 1, pp. 271–273, Jan 1994.

Code-based crypto

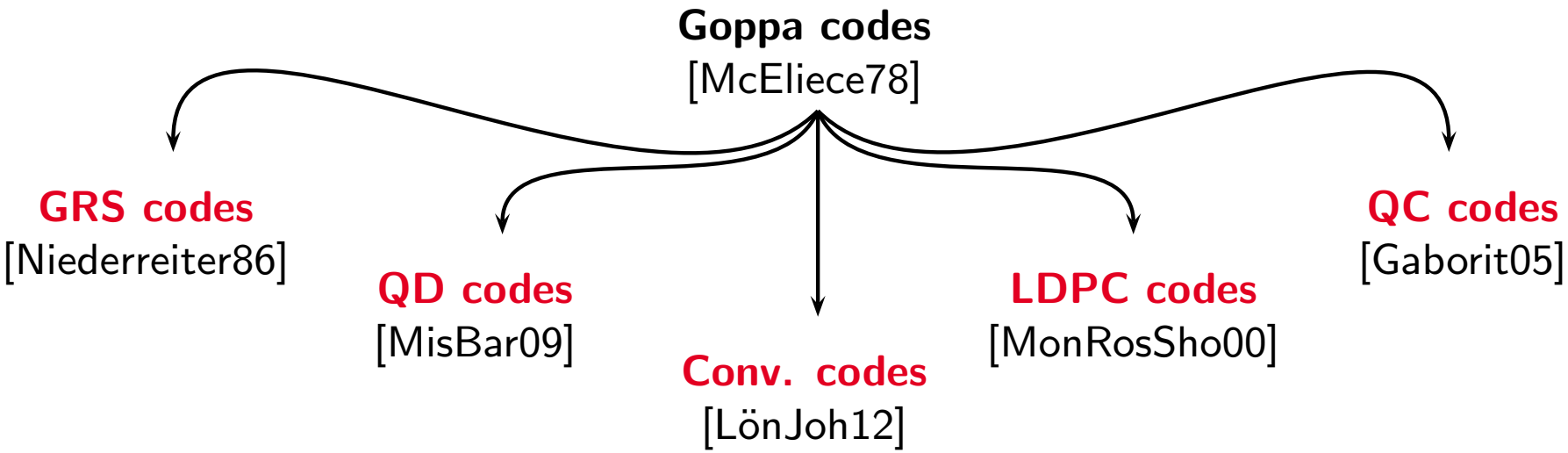
- Code-based public-key cryptosystems were introduced by McEliece in 1978, exploiting Goppa codes.
- Besides quantum resistant, they are algorithmically efficient.
- In 1986 Niederreiter introduced a variant in the syndrome domain, while McEliece works in the codeword domain.
- McEliece and Niederreiter indeed are two formulations of the same code-based trapdoor.
- Goppa codes have resisted cryptanalysis for 40 years...
- ...but they are large to store and slow to decode.

- R. McEliece, "Public-Key System Based on Algebraic Coding Theory," DSN Progress Report 44, pp. 114–116, 1978.
- H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," Problems of Control and Information Theory, vol. 15, pp. 159–166, 1986.
- Y. X. Li, R. H. Deng and X. M. Wang, "On the equivalence of McEliece's and Niederreiter's public-key cryptosystems," IEEE Trans. Inf. Theory, vol. 40, no. 1, pp. 271–273, Jan 1994.

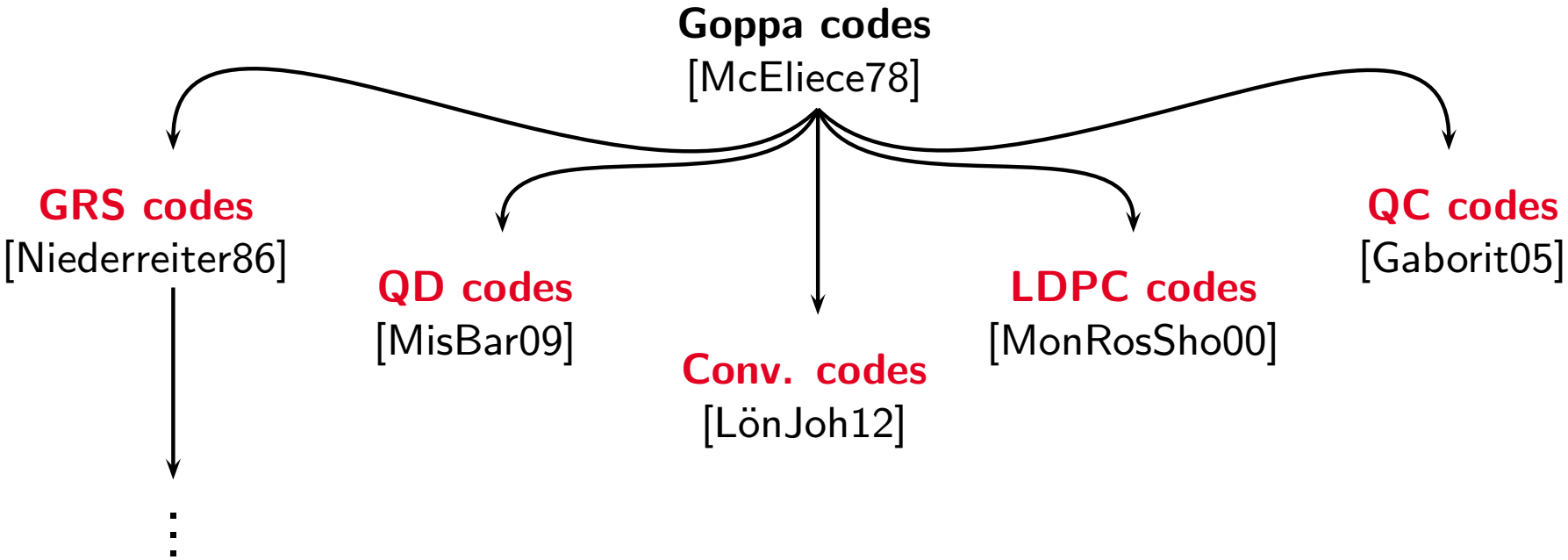
Goppa code replacements (in the Hamming metric)

Goppa codes
[McEliece78]

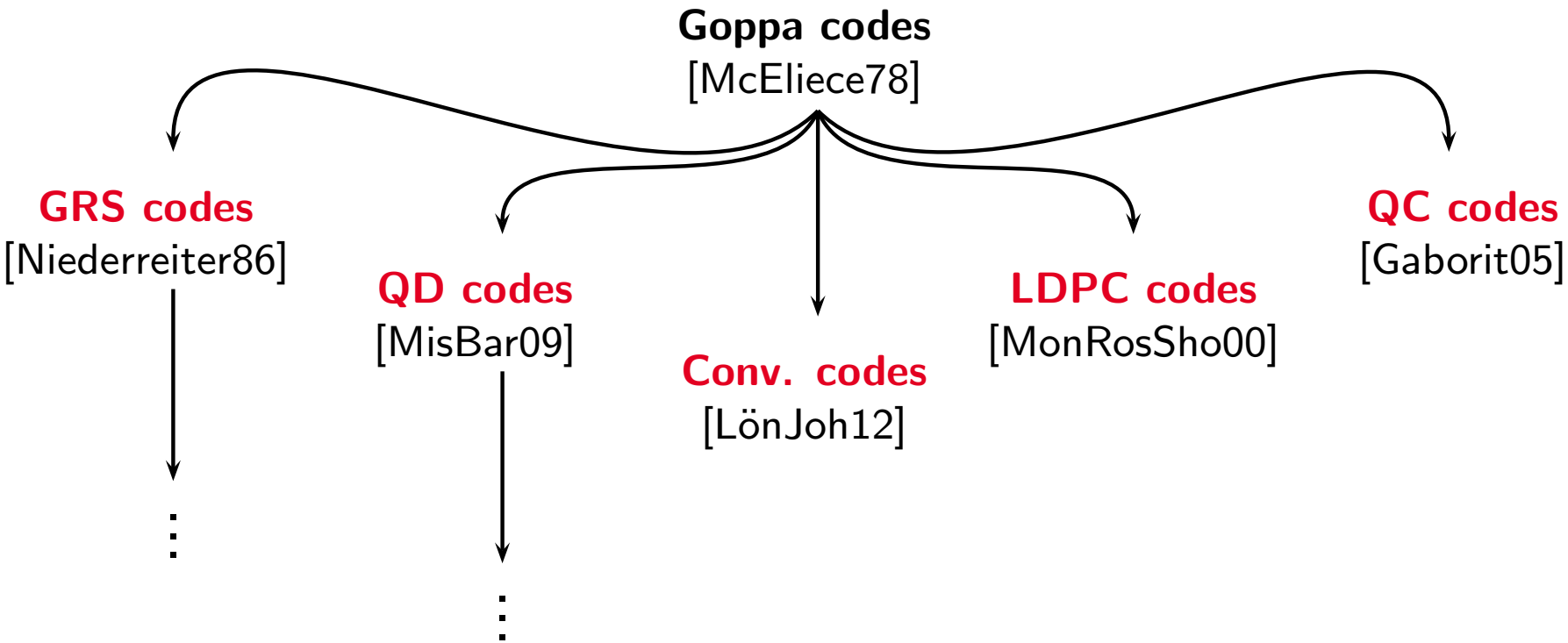
Goppa code replacements (in the Hamming metric)



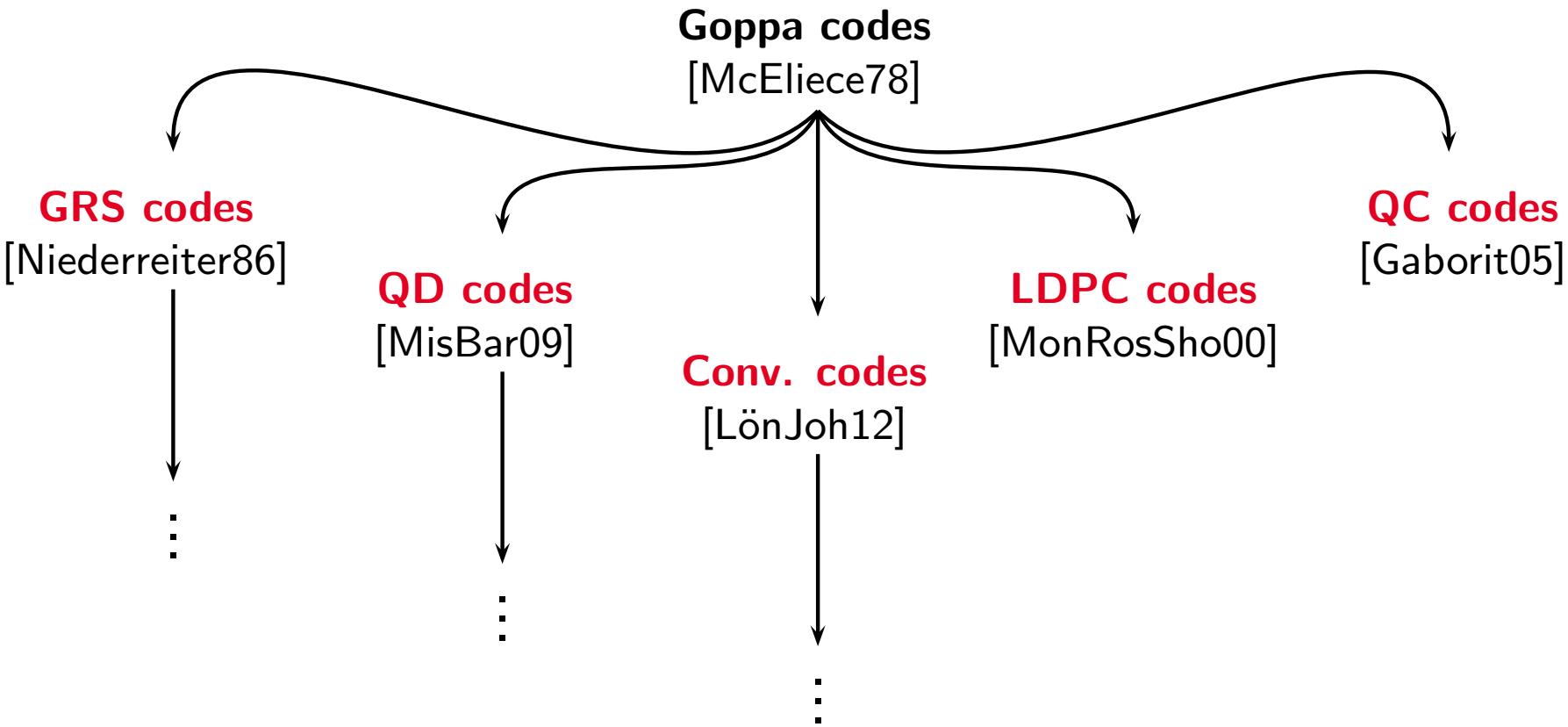
Goppa code replacements (in the Hamming metric)



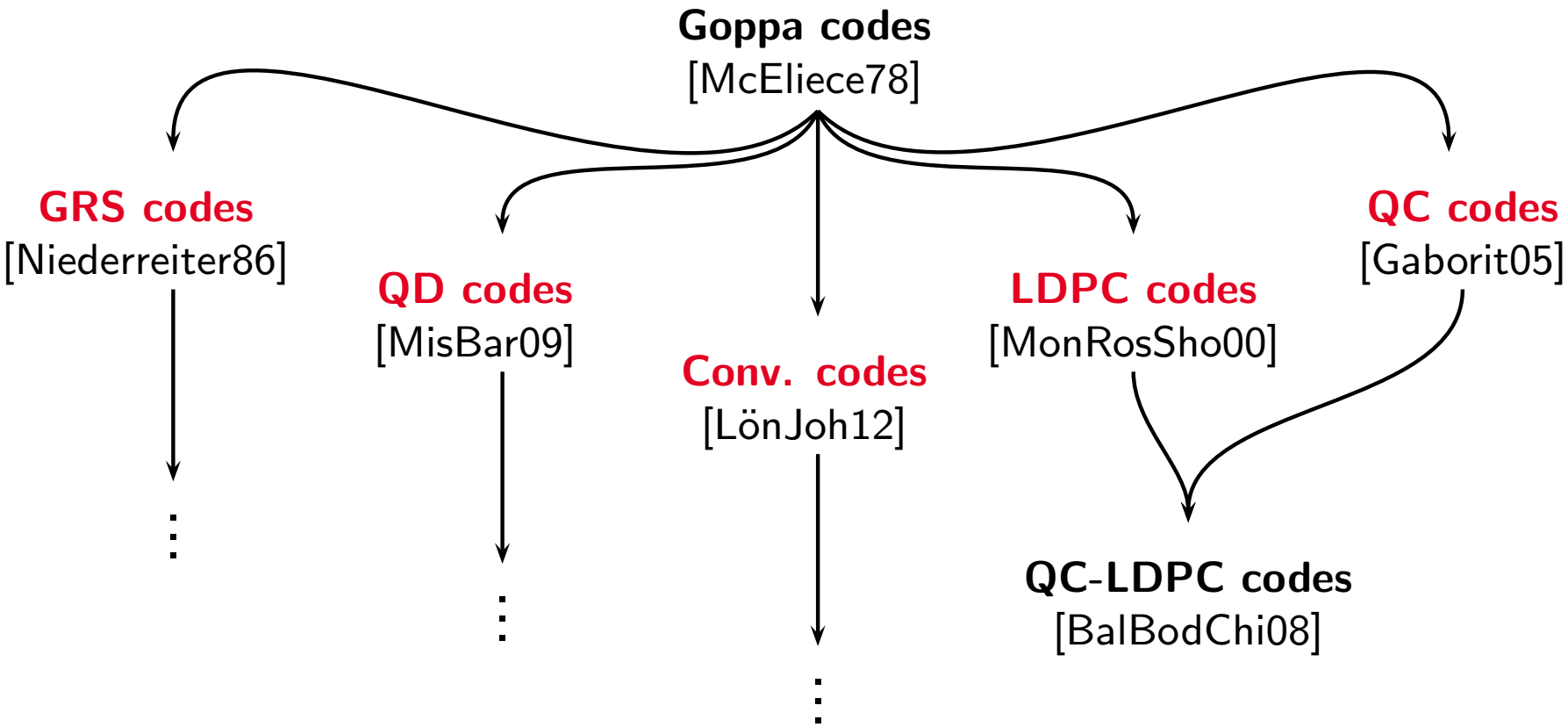
Goppa code replacements (in the Hamming metric)



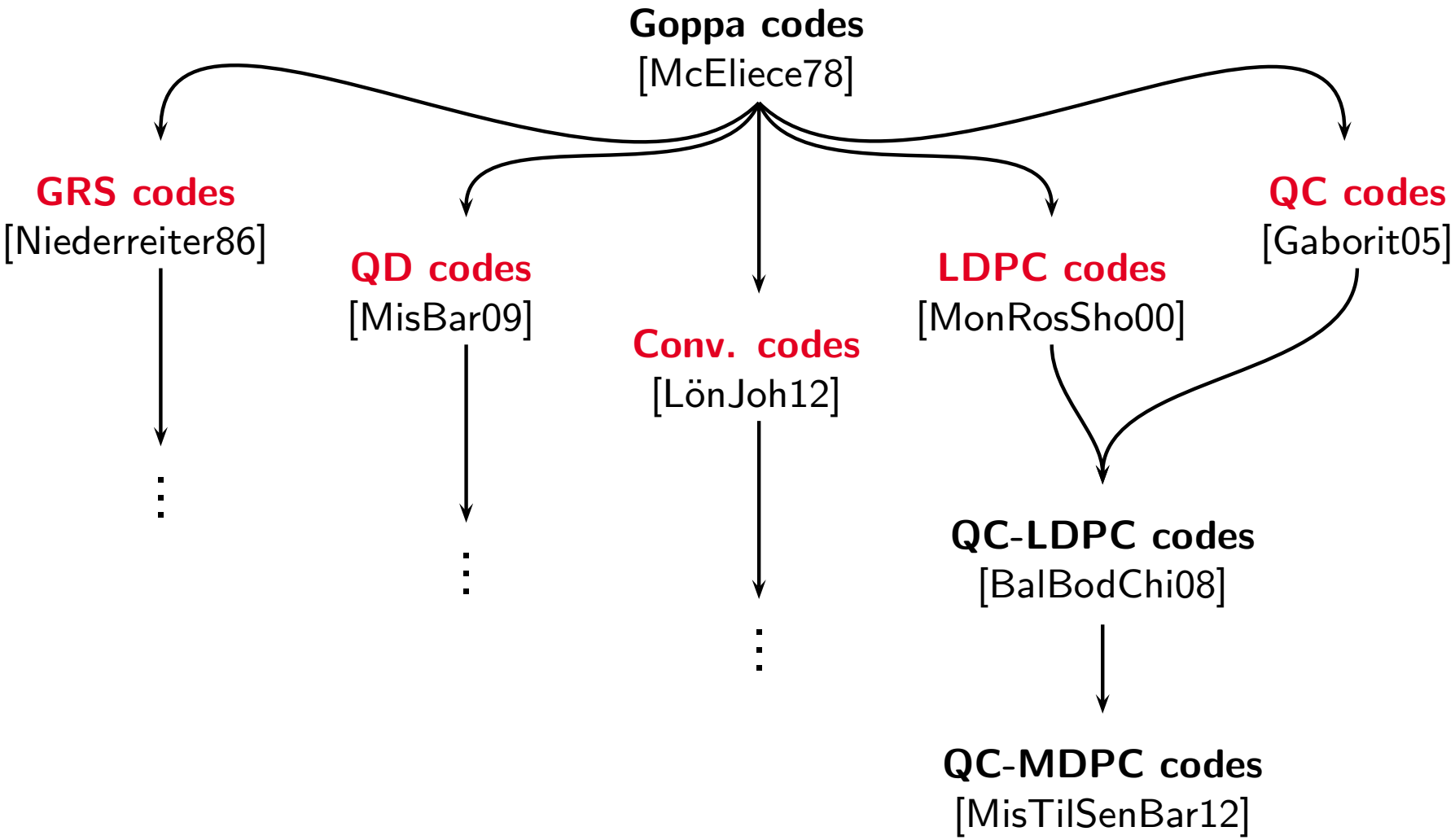
Goppa code replacements (in the Hamming metric)



Goppa code replacements (in the Hamming metric)



Goppa code replacements (in the Hamming metric)



LDPC codes in the McEliece cryptosystem

- Low-density parity-check (LDPC) codes are capacity-achieving codes under belief propagation (BP) decoding.
- They allow a random-based design, which results in large key spaces.
- The low density of their matrices could be attractive to achieve compact representations.
- All these makes them interesting for the use in McEliece/Niederreiter.

- C. Monico, J. Rosenthal, and A. Shokrollahi, "Using low density parity check codes in the McEliece cryptosystem," Proc. IEEE ISIT 2000, Sorrento, Italy, Jun. 2000, p. 215.
- M. Baldi, F. Chiaraluce, "Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes," Proc. IEEE ISIT 2007, Nice, France, Jun. 2007, pp. 2591–2595.

LDPC codes in the McEliece cryptosystem

- LDPC codes are capacity-achieving codes under BP decoding.
- They allow a random-based design, which results in large key spaces.
- The low density of their matrices could be attractive to achieve compact representations.
- All these makes them interesting for the use in McEliece/Niederreiter.

Warning

Public codes cannot be LDPC codes as well, otherwise secret codes are likely to be exposed.

- C. Monico, J. Rosenthal, and A. Shokrollahi, "Using low density parity check codes in the McEliece cryptosystem," Proc. IEEE ISIT 2000, Sorrento, Italy, Jun. 2000, p. 215.
- M. Baldi, F. Chiaraluce, "Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes," Proc. IEEE ISIT 2007, Nice, France, Jun. 2007, pp. 2591–2595.
- A. Otmani, J.P. Tillich, L. Dallot, "Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes," Proc. SCC 2008, Beijing, China, Apr. 2008.

LDPC codes in the McEliece cryptosystem (2)

Solutions

1. Use a transformation matrix $\mathbf{Q}$ that hides the sparse parity-check matrix $\mathbf{H}$ of the private code into a denser parity-check matrix $\mathbf{L} = \mathbf{H} \cdot \mathbf{Q}$ of the public code [BBC2008].
2. Use a private code defined by a denser parity-check matrix, called moderate-density parity-check (MDPC) matrix, and a permutation-equivalent public code [MTSB2013].

Pros and cons of 1 :

- Iterative decoding is more efficient on more sparse matrices.
- Multiplication by $\mathbf{Q}$ increases the weight of error vectors to be corrected during decryption.

- M. Baldi, M. Bodrato, F. Chiaraluce, "A new analysis of the McEliece cryptosystem based on QC-LDPC codes", Proc. SCN 2008, vol. 5229 of LNCS, pp. 246–262, 2008.
- R. Misoczki, J.-P. Tillich, N. Sendrier, P.S.L.M. Barreto, "MDPC-McEliece: new McEliece variants from moderate density parity-check codes", Proc. IEEE ISIT 2013, pp. 2069–2073, July 2013.

Example of QC-(almost)LDPC code

$$\mathbf{H} = \left[\begin{array}{ccccccccccc|ccccccccccc}
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1
\end{array}\right]$$

- Number of ciculant blocks: $n_0 = 2$.
- Code rate: $R = \frac{n_0 - 1}{n_0} = 1/2$.
- Parity-check matrix column weight: $d_v = 3$.
- Parity-check matrix row weight: $d_c = n_0 d_v = 6$.

QC-LDPC and QC-MDPC codes

$$\mathbf{H} = \left[\begin{array}{cccccccccc|cccccccccc} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ & & & & & \vdots & & & & & & & & & & \vdots & & & & & & \end{array}\right]$$

- The parity-check matrix is described by its first row.
- The storage size increases linearly in the code length.
- The code length is usually very large ($10,000 \lessapprox n \lessapprox 100,000$)
- $\mathbf{H}$ defines a QC-LDPC code if d_v is $\approx \log(n)$ ($d_v \lessapprox 20$).
- $\mathbf{H}$ defines a QC-MDPC code if d_v is $\approx \sqrt{n}$ ($50 \lessapprox d_v \lessapprox 150$).

Bit flipping decoding

- Classical hard-decision iterative decoding algorithms for LDPC codes are known as bit flipping (BF) algorithms.
- They are well suited when soft information from the channel is not available (as in the McEliece cryptosystem).
- From [Gallager1962]:

> The decoder computes all the parity checks and then changes any digit that is contained in more than some fixed number of unsatisfied parity-check equations. Using these new values, the parity checks are recomputed, and the process is repeated until the parity checks are all satisfied.

▶ R. G. Gallager, "Low-density parity-check codes," IRE Trans. Inform. Theory, vol. 8, pp. 21–28, 1962.

Contribution

1

Propose a *Low dEnsity coDe-bAsed key encapsulation mechanism* (LEDAkem).

Contribution

1

Propose a *Low dEnsity coDe-bAsed key encapsulation mechanism* (LEDAkem).

2

Propose an ad-hoc decoding algorithm exploiting the knowledge of $\mathbf{Q}$ to achieve better performance.

Contribution

1

Propose a *Low dEnsity coDe-bAsed key encapsulation mechanism* (LEDAkem).

2

Propose an ad-hoc decoding algorithm exploiting the knowledge of $\mathbf{Q}$ to achieve better performance.

3

Propose a reference and portable C99 implementation of LEDAkem.

LEDAkem functions

Key Generation

1. Generate a random $r \times n$ binary block circulant matrix $\mathbf{H} = [\mathbf{H}_0, \ldots, \mathbf{H}_{n_0-1}]$ with column weight $d_v \ll n$
2. Generate a random, non-singular, $n \times n$ binary block circulant matrix $\mathbf{Q}$ with column weight $m \ll n$
3. Compute $\mathbf{L} = \mathbf{H} \cdot \mathbf{Q} = [\mathbf{L}_0, \ldots, \mathbf{L}_{n_0-1}]$
4. Private key: $\mathbf{H}, \mathbf{Q}$; Public Key: $\mathbf{M} = (\mathbf{L}_{n_0-1})^{-1} \cdot \mathbf{L}$

LEDAkem functions (2)

Key Encapsulation

1. Generate a random n-bit error vector $\mathbf{e}$ with weight t
2. Compute the ciphertext (syndrome) $\mathbf{s} = \mathbf{M}\mathbf{e}^T$
3. Derive the shared secret $\mathbf{x} = \mathrm{KDF}(\mathbf{e})$

Key Decapsulation

1. Obtain $\mathbf{e}$ as $\textsc{Decode}(\mathbf{s}, \mathbf{H}, \mathbf{Q})$
2. Derive the shared secret $\mathbf{x} = \mathrm{KDF}(\mathbf{e})$

General attacks against McEliece/Niederreiter

Decoding attacks

Aimed at decrypting one or more ciphertexts without knowing the private key.

Key recovery attacks

Aimed at recovering the private key from the public key.

Decoding attacks

- The most dangerous decoding attacks (DAs) exploit information set decoding (ISD).
- The ISD principle was introduced by Prange in 1962.
- The first efficient algorithms were introduced by Lee-Brickell and Leon-Stern in 1988/89.
- These techniques have known great advancements in recent years.

- E. Prange, "The use of information sets in decoding cyclic codes, Information Theory," IRE Transactions on, vol. 8, no. 5, pp. 5–9, 1962.
- P. Lee, E. Brickell, "An observation on the security of McElieces public-key cryptosystem," Advances in Cryptology - EUROCRYPT 88, pp 275–280, 1988.
- J. Leon, "A probabilistic algorithm for computing minimum weights of large error-correcting codes," IEEE Trans. Inform. Theory, vol. 34, no. 5, pp. 1354–1359, 1988.
- J. Stern, "A method for finding codewords of small weight," Coding Theory and Applications, vol. 388 of Springer LNCS, pp. 106-113, 1989.

Modern information set decoding

- The general decoding problem can be reduced to that of searching low weight codewords.
- Modern approaches exploit the birthday paradox to search for low weight codewords.
- Lower bounds on complexity have been found recently by Niebuhr et al.

- C. Peters, "Information-set decoding for linear codes over F_q," Post-Quantum Cryptography, vol. 6061 of Springer LNCS, pp. 81–94, 2010.
- D. J. Bernstein, T. Lange, C. Peters, "Smaller decoding exponents: ball-collision decoding," CRYPTO 2011, vol. 6841 of Springer LNCS, pp 743–760, 2011.
- A. May, A. Meurer, E. Thomae, "Decoding random linear codes in $O(2^{0.054n})$," ASIACRYPT 2011, vol. 7073 of Springer LNCS, pp. 107124, 2011.
- A. Becker, A. Joux, A. May, and A. Meurer, "Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding," Advances in Cryptology - EUROCRYPT 2012, vol. 7237 of Springer LNCS, pp. 520–536, 2012.
- R. Niebuhr, E. Persichetti, P.-L. Cayrel, S. Bulygin, J. Buchmann, "On lower bounds for information set decoding over F_q and on the effect of partial knowledge," Int. J. Inf. Coding Theory, vol. 4, no. 1, pp. 47–78, 2017.

Pre-quantum VS post-quantum decoding attacks

- Grover's algorithm is a quantum algorithm introduced for finding all the roots of a Boolean function $\{0,1\}^n \rightarrow \{0,1\}$ in $O(\sqrt{2^n})$ instead of $O(2^n)$.

▶ S.H.S. de Vries, "Achieving 128-bit Security against Quantum Attacks in OpenVPN," Master Thesis, University of Twente, 2016.

Pre-quantum VS post-quantum decoding attacks

- Grover's algorithm is a quantum algorithm introduced for finding all the roots of a Boolean function $\{0,1\}^n \to \{0,1\}$ in $O(\sqrt{2^n})$ instead of $O(2^n)$.
- For searching one entry of an unsorted list of n entries,
  - The best classical algorithm requires $n/2$ steps on average.
  - Grover's algorithm requires $\pi/4\sqrt{n}$ steps using $\log_2(n)$ qubits.

▶ S.H.S. de Vries, "Achieving 128-bit Security against Quantum Attacks in OpenVPN," Master Thesis, University of Twente, 2016.

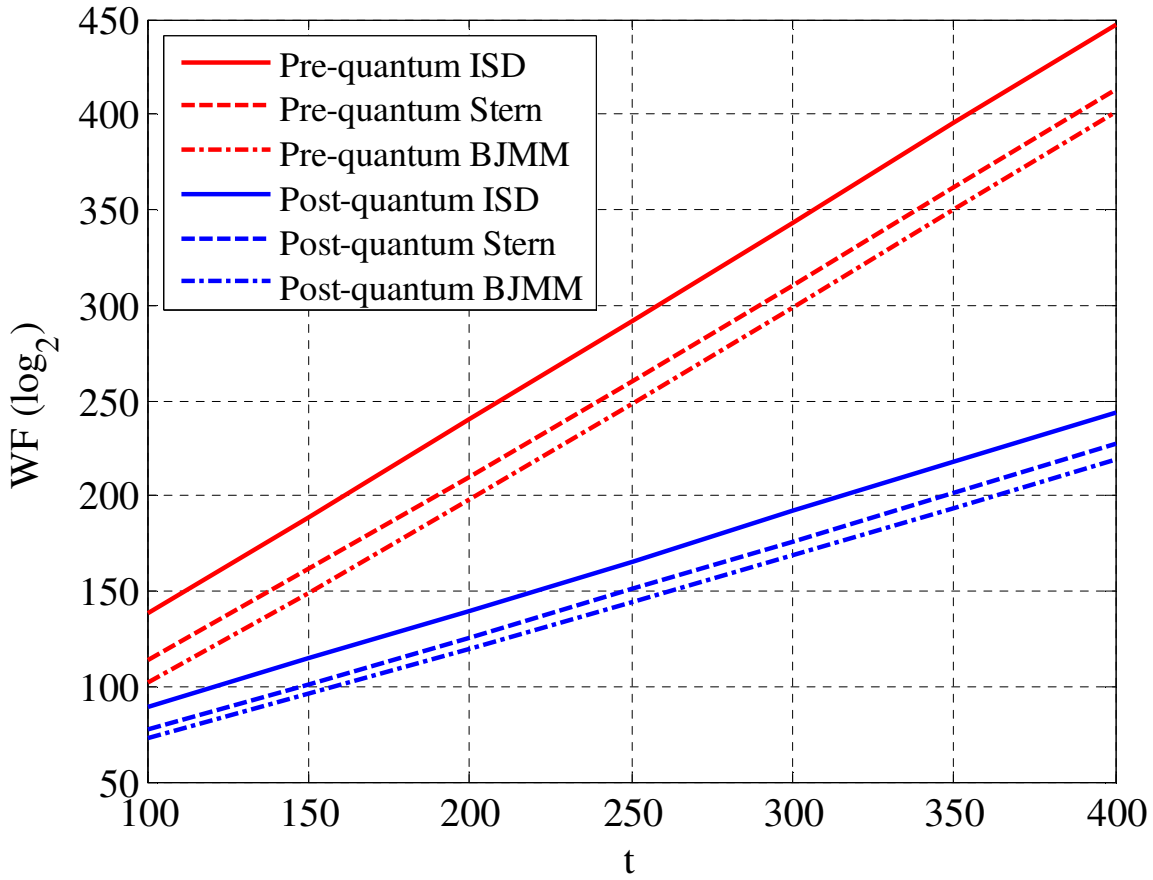
Pre-quantum VS post-quantum decoding attacks

- Grover's algorithm is a quantum algorithm introduced for finding all the roots of a Boolean function $\{0,1\}^n \to \{0,1\}$ in $O(\sqrt{2^n})$ instead of $O(2^n)$.
- For searching one entry of an unsorted list of n entries,
  - The best classical algorithm requires $n/2$ steps on average.
  - Grover's algorithm requires $\pi/4\sqrt{n}$ steps using $\log_2(n)$ qubits.
- Grover's algorithm reduces the number of iterations but does not reduce the cost per iteration.
- However, it somehow impacts the work factor of ISD.

▶ S.H.S. de Vries, "Achieving 128-bit Security against Quantum Attacks in OpenVPN," Master Thesis, University of Twente, 2016.

Pre-quantum VS post-quantum decoding attacks

Pre- and post-quantum WF of some ISD algorithms versus t, for codes with $n = 12000$, $k = 6000$.



Key recovery attacks based on ISD

- The matrix $\mathbf{L}$, with row weight $\leq n_0 m d_v$, is a valid parity-check matrix for the public code.
- An ISD algorithm can be used to search for the rows of $\mathbf{L}$ in the dual of the public code.

Key recovery attacks based on ISD

- The matrix $\mathbf{L}$, with row weight $\leq n_0 m d_v$, is a valid parity-check matrix for the public code.
- An ISD algorithm can be used to search for the rows of $\mathbf{L}$ in the dual of the public code.

Work Factor of ISD-based attacks

$$\mathrm{WF}_{\mathrm{DA}} = \frac{\mathrm{C}_{\mathrm{ISD}}(n, k, t)}{\sqrt{p}}, \quad \mathrm{WF}_{\mathrm{KRA}} = \frac{\mathrm{C}_{\mathrm{ISD}}(n, n-k, n_0 m d_v)}{p}$$

▶ N. Sendrier, "Decoding one out of many," in Proc. PQCrypto 2011, vol. 7071 of Springer LNCS, pp. 51–67, 2011.

Key recovery attacks based on decoding errors

- Recently, it has been shown that QC-MDPC and QC-LDPC code-based McEliece cryptosystem may suffer from reaction attacks exploiting decoding errors.

- Q. Guo, T. Johansson, P. Stankovski, "A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors", ASIACRYPT 2016, vol. 10031 of Springer LNCS, pp. 789–815.
- T. Fabšič, V. Hromada, P. Stankovski, P. Zajac, Q. Guo, T. Johansson, "A Reaction Attack on the QC-LDPC McEliece Cryptosystem", PQCrypto 2017, vol. 10346 of Springer LNCS, pp. 51–68.
- T. Fabšič, V. Hromada, P. Stankovski, "A Reaction Attack on LEDApkc", Cryptology ePrint Archive, Report 2018/140, 2018.

Key recovery attacks based on decoding errors

- Recently, it has been shown that QC-MDPC and QC-LDPC code-based McEliece cryptosystem may suffer from reaction attacks exploiting decoding errors.
- The attack is built upon two facts:
  1. The decryption failure probability is non-zero and depends on the structure of the secret key.
  2. Eve can estimate such a probability by observing Bob's reactions during decryption of some special ciphertexts.

- Q. Guo, T. Johansson, P. Stankovski, "A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors", ASIACRYPT 2016, vol. 10031 of Springer LNCS, pp. 789–815.
- T. Fabšič, V. Hromada, P. Stankovski, P. Zajac, Q. Guo, T. Johansson, "A Reaction Attack on the QC-LDPC McEliece Cryptosystem", PQCrypto 2017, vol. 10346 of Springer LNCS, pp. 51–68.
- T. Fabšič, V. Hromada, P. Stankovski, "A Reaction Attack on LEDApkc", Cryptology ePrint Archive, Report 2018/140, 2018.

Key recovery attacks based on decoding errors

- Recently, it has been shown that QC-MDPC and QC-LDPC code-based McEliece cryptosystem may suffer from reaction attacks exploiting decoding errors.
- The attack is built upon two facts:
  1. The decryption failure probability is non-zero and depends on the structure of the secret key.
  2. Eve can estimate such a probability by observing Bob's reactions during decryption of some special ciphertexts.
- LEDAkem thwarts these attacks by using ephemeral (i.e., one-time) keypairs.

- Q. Guo, T. Johansson, P. Stankovski, "A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors", ASIACRYPT 2016, vol. 10031 of Springer LNCS, pp. 789–815.
- T. Fabšič, V. Hromada, P. Stankovski, P. Zajac, Q. Guo, T. Johansson, "A Reaction Attack on the QC-LDPC McEliece Cryptosystem", PQCrypto 2017, vol. 10346 of Springer LNCS, pp. 51–68.
- T. Fabšič, V. Hromada, P. Stankovski, "A Reaction Attack on LEDApkc", Cryptology ePrint Archive, Report 2018/140, 2018.

Rationale of the Q-decoder

- Decoding is performed on the syndrome

$$\mathbf{s} = \mathbf{e} \cdot \mathbf{L}^T = \mathbf{e} \cdot \mathbf{Q}^T \cdot \mathbf{H}^T = \mathbf{e}' \cdot \mathbf{H}^T$$

  where $\mathbf{e}' = \mathbf{e} \cdot \mathbf{Q}^T$ is the expanded error vector.
- Let $\phi(\mathbf{e})$ denote the support of $\mathbf{e}$ and $\mathbf{q}_j$ be the j-th row of $\mathbf{Q}^T$, then

$$\mathbf{e}' = \sum_{j \in \phi(\mathbf{e})} \mathbf{q}_j$$

Rationale of the Q-decoder

- Decoding is performed on the syndrome

$$\mathbf{s} = \mathbf{e} \cdot \mathbf{L}^T = \mathbf{e} \cdot \mathbf{Q}^T \cdot \mathbf{H}^T = \mathbf{e}' \cdot \mathbf{H}^T$$

  where $\mathbf{e}' = \mathbf{e} \cdot \mathbf{Q}^T$ is the expanded error vector.
- Let $\phi(\mathbf{e})$ denote the support of $\mathbf{e}$ and $\mathbf{q}_j$ be the j-th row of $\mathbf{Q}^T$, then

$$\mathbf{e}' = \sum_{j \in \phi(\mathbf{e})} \mathbf{q}_j$$

- The rows of $\mathbf{Q}^T$ are sparse ($\mathrm{wt}(\mathbf{q}_i) = m \ll n$), hence their supports are (almost) disjoint.

Rationale of the Q-decoder

- Decoding is performed on the syndrome

$$\mathbf{s} = \mathbf{e} \cdot \mathbf{L}^T = \mathbf{e} \cdot \mathbf{Q}^T \cdot \mathbf{H}^T = \mathbf{e}' \cdot \mathbf{H}^T$$

  where $\mathbf{e}' = \mathbf{e} \cdot \mathbf{Q}^T$ is the expanded error vector.
- Let $\phi(\mathbf{e})$ denote the support of $\mathbf{e}$ and $\mathbf{q}_j$ be the j-th row of $\mathbf{Q}^T$, then

$$\mathbf{e}' = \sum_{j \in \phi(\mathbf{e})} \mathbf{q}_j$$

- The rows of $\mathbf{Q}^T$ are sparse ($\mathrm{wt}(\mathbf{q}_i) = m \ll n$), hence their supports are (almost) disjoint.
- Also $\mathbf{e}$ is sparse ($\mathrm{wt}(\mathbf{e}) = t \ll n$), hence

$$\mathrm{wt}(\mathbf{e}') \approx mt$$

Rationale of the Q-decoder (2)

- Let us consider the (integer) inner product $\rho = \mathbf{e}' * \mathbf{q}_v$:
  - if $v \notin \phi(\mathbf{e})$, then the supports of $\mathbf{e}'$ and $\mathbf{q}_v$ have a small intersection and ρ is small;
  - if $v \in \phi(\mathbf{e})$, then $\mathbf{q}_v$ is one of the rows forming $\mathbf{e}'$, hence ρ is large.

Rationale of the Q-decoder (2)

- Let us consider the (integer) inner product $\rho = \mathbf{e}' * \mathbf{q}_v$:
  - if $v \notin \phi(\mathbf{e})$, then the supports of $\mathbf{e}'$ and $\mathbf{q}_v$ have a small intersection and ρ is small;
  - if $v \in \phi(\mathbf{e})$, then $\mathbf{q}_v$ is one of the rows forming $\mathbf{e}'$, hence ρ is large.
- As in BF decoding, an estimate of $\mathbf{e}'$ is obtained by computing the (integer) inner product between the syndrome and each column of $\mathbf{H}$

$$\mathbf{\Sigma} = \mathbf{s} * \mathbf{H}$$

and thresholding the vector $\mathbf{\Sigma}$.

Rationale of the Q-decoder (2)

- Let us consider the (integer) inner product $\rho = \mathbf{e}' * \mathbf{q}_v$:
  - if $v \notin \phi(\mathbf{e})$, then the supports of $\mathbf{e}'$ and $\mathbf{q}_v$ have a small intersection and ρ is small;
  - if $v \in \phi(\mathbf{e})$, then $\mathbf{q}_v$ is one of the rows forming $\mathbf{e}'$, hence ρ is large.
- As in BF decoding, an estimate of $\mathbf{e}'$ is obtained by computing the (integer) inner product between the syndrome and each column of $\mathbf{H}$

$$\mathbf{\Sigma} = \mathbf{s} * \mathbf{H}$$

  and thresholding the vector $\mathbf{\Sigma}$.
- So we can estimate $\phi(\mathbf{e})$ by replacing $\mathbf{e}'$ with $\mathbf{\Sigma}$ to compute

$$\mathbf{R} = [\rho_0, \rho_1, \cdots, \rho_{n-1}] = \mathbf{\Sigma} * \mathbf{Q}$$

  and thresholding the vector $\mathbf{R}$.

Q-decoder

Initialization

$\mathbf{e} = \mathbf{0}_{1\times n}, \quad \mathbf{s} = \mathbf{e}\cdot\mathbf{L}^T.$

Description of the j-th iteration

Input: $\mathbf{e}^{(j-1)}, \mathbf{s}^{(j-1)}$

1. Compute $\mathbf{\Sigma} = [\sigma_0, \sigma_1, \cdots, \sigma_{n-1}] = \mathbf{s}^{(j-1)} * \mathbf{H}$.
2. Compute $\mathbf{R} = [\rho_0, \rho_1, \cdots, \rho_{n-1}] = \mathbf{\Sigma} * \mathbf{Q}$.
3. Compute $\Psi = \left\{ i \,\middle|\, \rho_i \geq b^{(j)} \right\}$.
4. Update the error vector as $\mathbf{e}^{(j)} = \mathbf{e}^{(j-1)} + \mathbf{1}_\Psi$.
5. Update the syndrome as $\mathbf{s}^{(j)} = \mathbf{s}^{(j-1)} + \sum_{i\in\Psi} \mathbf{q}_i \cdot \mathbf{H}^T$.

Output: $\mathbf{e}^{(j)}, \mathbf{s}^{(j)}$

Choice of the flipping thresholds

- Through combinatorial arguments (details in the paper) we can estimate $P\{e_i = 1 | \rho_i\}$.

- J. Chaulet and N. Sendrier, "Worst case QC-MDPC decoder for McEliece cryptosystem," Proc. IEEE ISIT 2016, Barcelona, 2016, pp. 1366–1370.

Choice of the flipping thresholds

- Through combinatorial arguments (details in the paper) we can estimate $P\{e_i = 1|\rho_i\}$.
- Let us define a decision margin $\Delta \geq 0$ and consider the decision condition

$$P\{e_i = 1|\rho_i\} > (1+\Delta)P\{e_i = 0|\rho_i\} = \frac{1+\Delta}{2+\Delta},$$

since $P\left\{e_i = 0|\rho_i^{(l)}\right\} = 1 - P\left\{e_i = 1|\rho_i^{(l)}\right\}$.

- Increasing Δ increases the average number of iterations, but reduces the DFR.

▶ J. Chaulet and N. Sendrier, "Worst case QC-MDPC decoder for McEliece cryptosystem," Proc. IEEE ISIT 2016, Barcelona, 2016, pp. 1366–1370.

Choice of the flipping thresholds

- Through combinatorial arguments (details in the paper) we can estimate $P\{e_i = 1|\rho_i\}$.
- Let us define a decision margin $\Delta \geq 0$ and consider the decision condition

$$P\{e_i = 1|\rho_i\} > (1+\Delta)P\{e_i = 0|\rho_i\} = \frac{1+\Delta}{2+\Delta},$$

since $P\left\{e_i = 0|\rho_i^{(l)}\right\} = 1 - P\left\{e_i = 1|\rho_i^{(l)}\right\}$.

- Increasing Δ increases the average number of iterations, but reduces the DFR.
- The corresponding decision threshold value is

$$b = \min\left\{\rho_i \in [0; md_v], \text{ s.t. } P\{e_i = 1|\rho_i\} > \frac{1+\Delta}{2+\Delta}\right\}$$

▶ J. Chaulet and N. Sendrier, "Worst case QC-MDPC decoder for McEliece cryptosystem," Proc. IEEE ISIT 2016, Barcelona, 2016, pp. 1366–1370.

Choice of the flipping thresholds (2)

- $P\{e_i = 1|\rho_i\}$ depends on the weight of the error vector.
- The weight of the error vector can be related to the syndrome weight (w).
- We can exploit a look-up table populated with the pairs $\{w, b\}$, sequentially ordered.
- During the l-th iteration:
  1. compute the syndrome weight $w^{(l)}$,
  2. search the largest w in the look-up table such that $w < w^{(l)}$,
  3. set $b^{(l)} = b$.

Proposed parameters

Table: Parameters for LEDAkem and estimated computational efforts to break a given instance as a function of the security category and number of circulant blocks n_0

Category	n_0	p	d_v	$[m_0, \cdots, m_{n_0-1}]$	t	$SL_{DA}^{(pq)}$	$SL_{KRA}^{(pq)}$	$SL_{DA}^{(cl)}$	$SL_{KRA}^{(cl)}$	DFR
1	2	27,779	17	[4, 3]	224	135.43	134.84	217.45	223.66	$\approx 8.3 \cdot 10^{-9}$
	3	18,701	19	[3, 2, 2]	141	135.63	133.06	216.42	219.84	$\lesssim 10^{-9}$
	4	17,027	21	[4, 1, 1, 1]	112	136.11	139.29	216.86	230.61	$\lesssim 10^{-9}$
2–3	2	57,557	17	[6, 5]	349	200.47	204.84	341.52	358.16	$\lesssim 10^{-8}$
	3	41,507	19	[3, 4, 4]	220	200.44	200.95	341.61	351.57	$\lesssim 10^{-8}$
	4	35,027	17	[4, 3, 3, 3]	175	200.41	201.40	343.36	351.96	$\lesssim 10^{-8}$
4–5	2	99,053	19	[7, 6]	474	265.38	267.00	467.24	478.67	$\lesssim 10^{-8}$
	3	72,019	19	[7, 4, 4]	301	265.70	270.18	471.67	484.48	$\lesssim 10^{-8}$
	4	60,509	23	[4, 3, 3, 3]	239	265.48	268.03	473.38	480.73	$\lesssim 10^{-8}$

- Public key size $= (n_0 - 1)p$.
- Ranging between 3 and 22 KiB.

Efficient implementation

Circulant matrix representation/arithmetics

- Represent circulant blocks as elements of $\mathbb{F}_2[x]/\langle x^p + 1\rangle$
  - Reduces both time and space complexity for arithmetics

Remove invertibility check for $\mathbf{Q}$

- Perm($\mathbf{Q}$) is odd and $< p \Rightarrow \mathbf{Q}$ is invertible

Efficient implementation

Circulant matrix representation/arithmetics

- Represent circulant blocks as elements of $\mathbb{F}_2[x]/\langle x^p + 1\rangle$
  - Reduces both time and space complexity for arithmetics

Remove invertibility check for $\mathbf{Q}$

- Perm($\mathbf{Q}$) is odd and $< p \Rightarrow \mathbf{Q}$ is invertible

- Reference implementation in ISO-C99 without platform-specific optimization.
  - NIST ref. platform: Base Intel x86-64 ISA (early 2006 CPUs).
- Possible further optimizations:
  - Sub-quadratic poly multiplication arithmetics.
  - Use x86-64 ISA extensions (e.g. CLMUL) and vector units.
  - Design a dedicated HW implementation.

Implementation results

Table: Running times for key generation, encryption and decryption as a function of the category and the number of circulant blocks n_0 on an AMD Ryzen 5 1600 CPU.

Category	n_0	KeyGen (ms)	Encrypt (ms)	Decrypt (ms)	Total CPU time Ephemeral KEM (ms)
1	2	34.11 ($\pm$1.07)	2.11 ($\pm$0.08)	16.78 ($\pm$0.53)	52.99
	3	16.02 ($\pm$0.26)	2.15 ($\pm$0.17)	21.65 ($\pm$1.71)	39.81
	4	13.41 ($\pm$0.23)	2.42 ($\pm$0.08)	24.31 ($\pm$0.86)	40.14
2–3	2	142.71 ($\pm$1.52)	8.11 ($\pm$0.21)	48.23 ($\pm$2.93)	199.05
	3	76.74 ($\pm$0.78)	8.79 ($\pm$0.20)	49.15 ($\pm$2.20)	134.68
	4	54.93 ($\pm$0.84)	9.46 ($\pm$0.28)	46.16 ($\pm$2.03)	110.55
4–5	2	427.38 ($\pm$5.15)	23.00 ($\pm$0.33)	91.78 ($\pm$5.38)	542.16
	3	227.71 ($\pm$1.71)	24.85 ($\pm$0.37)	92.42 ($\pm$4.50)	344.99
	4	162.34 ($\pm$2.39)	26.30 ($\pm$0.53)	127.16 ($\pm$4.42)	315.80

Thanks for the attention

Questions?

https://www.ledacrypt.org